

Bachelor's Thesis (UAS)

Degree Program: Bachelor's Degree

Specialization: Information Technology

2015

Ganga Lingden

REMOTE MANAGEMENT OF LIQUID LEVEL USING RASPBERRY PI WITH ULTRASONIC SENSOR



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree Programme in Information Technology | Information Technology

2015 | 62

Instructor: Granholm Patric

Ganga Dhvaj Lingden

REMOTE MANAGEMENT OF LIQUID LEVEL USING RASPBERRY PI WITH ULTRASONIC SENSOR

Modern system management has been used remotely using various tools and techniques. One such practical example is fuel level management at a gas station. Fuel levels can be managed with remote management, saving extra costs and manpower. This kind of remote management can be achieved a solution implemented with Raspberry Pi, ultrasonic sensors, and Arduino. Most specifically, ultrasonic sensors equip the Arduino in the container of the fuel to update the status of the current level of liquid as per the parameter of the system design, sending data to the Raspberry Pi-based system. Thus, this kind of solution helps to obtain timely updated information on the fuel level as well as lowers the need of day-to-day management service.

The main objective of this thesis work was to develop a remote liquid level measurement system with embedding Raspberry Pi, Arduino and Ultrasonic sensor, either using Wi-Fi Internet or Cellular Network.

Thus, with the use of low cost hardware devices, along with the modern programming language like C#, ASP.NET, HTML, CSS, Python, and the MySQL database, a very reliable system was generated to manage remotely the liquid level at a remote gas station.

KEYWORDS:

Raspberry Pi, Arduino Uno, Ultrasonic Sensor, Internet of Things, HTML, CSS, C#, ASP.NET, MySQL, Python.

CONTENTS

List of abbreviations (or) symbols

1 INTRODUCTION	9
2 ULTRASONIC SENOSR	11
2.1 Properties and Effect of External Factors.....	11
2.2 Importance and drawbacks of ultrasonic sensors.....	14
2.3 HC-SR04 Ultrasonic Module	14
2.3.1 Electric Parameters and Its Limitation:	15
2.3.2 Operation Principle.....	16
2.3.3 Practical Consideration.....	17
3 ARDUINO UNO	19
3.1 Arduino Technical Specification	20
3.2 Input and Output Operation.....	22
3.3 Power Supply Consideration.....	23
3.4 Arduino Software	23
4 RASPBERRY PI.....	25
4.1 Modular Structure and Specification	26
4.2 Power Supply Consideration.....	27
4.3 GPIO Hardware Connector	28
5 SOFTWARE TOOLS OVERVIEW.....	30
5.1 Raspbian	30
5.2 Python	30
5.3 MySQL server.....	31
5.4 VNC Server	31
5.5 IIS Express	32
5.6 Visual Studio.....	32
5.7 HTML.....	32
5.8 CSS	33
5.9 ASP.NET	33
5.10 C Sharp (C#).....	34

6 PROJECT DESIGN AND DEVELOPMENT	35
6.1 Project System Design.....	35
6.2 Remote Site Network Development	36
6.2.1 Arduino Modification with HC-SR04 Sensor	37
6.2.2 Raspberry Pi Modification.....	39
6.3 Arduino and Raspberry Pi Communication	41
6.4 Case Study based on Liquid Tank Structure	42
6.5 Central Monitoring Station.....	46
6.5.1 Database Connection	46
6.5.2 Secure Remote Connection	47
6.5.3 Web Interface	48
7 DATA SECURITY AND CHALLENGES.....	49
7.1 Network and Data Security	49
7.2 Challenges.....	50
8 CONCLUSION.....	51
REFERENCES	52

APPENDICES

Appendix 1.0 - Source code to setup the Arduino for HCSR-04 Sensor	54
Appendix 2.0 – Python script source code to retrieve the data from the Arduino to Raspberry Pi	55
Appendix 3.0 – Source code of HTML file	56
Appendix 4.0 – Source Code of CSS file (Style.css)	57
Appendix 5.0 - C# Source code to display data on the web page	60
Appendix 6.0 – Database Connection String	61
Appendix 7.0 – ASP.NET Source Code	62

LIST OF FIGURES

Figure 1. Wavelength of Sound Wave	11
Figure 2. Ultrasonic Wave Propagation	12
Figure 3. Attenuation characteristics of sound pressure with distance	13
Figure 4. Ultrasonic Sensor: HC-SR04	15
Figure 5. Ultrasonic HR-SR04 Module Timing Diagram [Ultrasonic Ranging Module HC-SR04, 2015]	17
Figure 6. Error Case in Object Position	18
Figure 7. Arduino Uno and its Component.....	20
Figure 8. Arduino IDE showing 'blink' program.....	24
Figure 9. Raspberry Pi Model B.....	26
Figure 10. GPIO Header in Model B Pi [Raspberry Pi, Wikipedia 2015]	29
Figure 11. Project System design	36
Figure 12. Remote Site Network Topology	37
Figure 13. Connection between HC-SR04 and Arduino Uno board	38
Figure 14. Screenshot of Arduino Serial Monitor Output	39
Figure 15. Static IP address Configuration	39
Figure 16. Raspberry Pi Desktop in VNC viewer.....	40
Figure 17. Manually Inserted data into Database	41
Figure 18. Liquid level measuring in Vertical Horizontal Cylindrical tank	43
Figure 19. Liquid level measuring in Horizontal Cylindrical Tank	44
Figure 20. Liquid level measuring in Rectangle Tank.....	44
Figure 21. Liquid level measuring in cube tank	45
Figure 22. Liquid level measuring in Horizontal Oval Tank	46
Figure 23. Screenshot of Database management in MySQL Workbench.....	47
Figure 24. Screenshot of data display on the web page.....	48

List of tables

Table 1. HC-SR04 Electric Parameters [Ultrasonic Ranging Module HC-SR04, 2015]	16
Table 2. Technical Specification of Arduino Uno. [Arduino Uno, 2015]	20
Table 3. Raspberry Pi Model A and B specification [Raspberry Pi, Element 14 2015]	27

List of abbreviations (or) symbols

PWM	Pulse Width Modulation
TTL	Transistor –transistor logic
AC	Alternating Current
DC	Direct Current
ICSP	In Circuit Serial Programming
IDE	Integrated Development Environment
SRAM	Static Random-Access Memory
EEPROM	Electrical Erasable Programmable Read-Only Memory
Hz	Hertz
KHz	Kilo Hertz
MHz	Mega Hertz
VIN	Voltage Input
GND	Ground
FTDI	Future Technology Devices International
TX	Transmitter
RX	Receiver
MB	Mega Bytes
SD	Secure Digital
CSS	Cascading Style Sheets
HTML	HyperText Markup Language

ASP.NET	Active Server Page
SQL	Structured Query Language
IIS	Internet Information Service
VNC	Virtual Network Computing
SSH	Secure Shell
HTTP	Hypertext Transfor Protocol
HTTPS	Secure Hypertext Transfor Protocol
FTP	File Transfor Protocol
SMPT	Small Mail Transfor Protocol
NNPT	Network News Transfor Protocol

1 INTRODUCTION

The network rises to full fame with implementation in varied fields, making the Internet as a structured backbone either via Wi-Fi or cellular network. The majority of complicated network has been proven asset to the logical and economic world. Integration of hardware and software give birth to various technologies, implementing a solution to real life problems. There are various examples such as embedded accounting machines, embedded hardware solutions and so on, which provide enough support to discuss Raspberry Pi application with integrated Arduino.

The idea of remote monitoring of liquid measurement level came up as a solution with the motive of initiating a solution to manage remote notification of liquid measurement level. In order to manage the idea, it has been tested to implement solution made from Raspberry Pi, Arduino, and Sensors etc. And, in order to expose it to scalability, it would be integrated into Wi-Fi Network or Cellular Network to connect it to more devices and provide transparent data for accurate notification. Besides that, it could be managed from the server with static Internet Protocol address, integrated with centralized management software. Thus, it helps to view and notifies the liquid level on time based and makes daily management easier and less time consuming.

Feasibility could also be a topic further discussed regarding the project. This project could be applied in real life solution like fuel station, to manage daily fuel level storage in the gas station, to manage daily water supply from a water provider. Apart from this, the project could be implemented to any other liquid industries, where the notification of liquid measurement level is needed. It has a wide range of application in economic as well as commercial ways, with less financial support, with greater commercial benefit.

The thesis work is split into 8 chapters. Chapter 1 refers to the introduction, motivation and objective of the work. In Chapter 2-4, the different hardware devices that have been used, and their features and limitations are described. The literature overview of used software applications and their implementation in the project are mentioned in the

chapter 5. In chapter 6, the whole system design and its development are presented, along with the modification of Arduino, Raspberry Pi and the Sensor. The challenges that have been encountered during the development of the project, and its security measures are identified in the chapter 7. Likewise, the conclusion, is presented in chapter 8.

2 ULTRASONIC SENOSR

Ultrasonic sensor is an electronic device that converts the electrical energy into the ultrasound, or sound waves above the normal range of the human being's hearing power or vice versa [ultrasonic transducer, 2015]. It is based on the principle that the sound waves have constant velocity. An ultrasonic sensor emits the waves into the air medium, and detects the reflected waves from the object, in order to determine the distance of an object. The time for ultrasonic waves to strike on the target object and to return back to the source is directly proportional to the distance to the object. Apart from the distance measurement, ultrasonic sensors are generally utilized for so many reasons, for example, material testing (to identify air bubbles, leakage and other flaws in the products), object identification, object's position discovery, ultrasonic mouse, and much more. The ultrasonic waves are the sounds having the frequencies above the human hearing range (20Hz - 20KHz), while the normal frequency of ultrasonic waves is above 20 KHz.

2.1 Properties and Effect of External Factors

In the section, the properties of ultrasonic wave and its effect due to external factors are discussed:

- Wavelengths and Radiation

Wavelength is defined as the distance between two successive crests or troughs. It is represented with the symbol ' λ '. The formula to calculate the wavelength is:

$\lambda = v/f$, where v is the velocity and f is the frequency of a sound wave.



Figure 1. Wavelength of Sound Wave

With a presence of the air at temperature 20 °C, the velocity of a ultrasonic sound wave is approximately 344 m/s. Due to the low velocity of the sound wave, the wavelength is short, which results, to get the higher resolution distance and direction. Thus, because of the higher resolution distance and direction of the waves, it is possible to get a large accuracy in the measurement. And, an accurate radiation can be easily achieved from the surface dimension of the ultrasonic sensor [Ultrasonic sensor, 2015].

- Reflection

It is one of the basic properties of ultrasonic wave on which the working mechanism of ultrasonic sensor is based on. The ultrasonic waves should be reflected back to the source in order to detect an object. Therefore, an object, which has a good reflectivity, is easy to detect. Some of the objects having 100% reflectivity of ultrasonic waves are metal, paper, glass, concrete, but it is difficult to identify the objects, which have large surface undulation because of the sporadic reflection, furthermore to the objects which absorb the waves, for example, clothes, wool, etc.

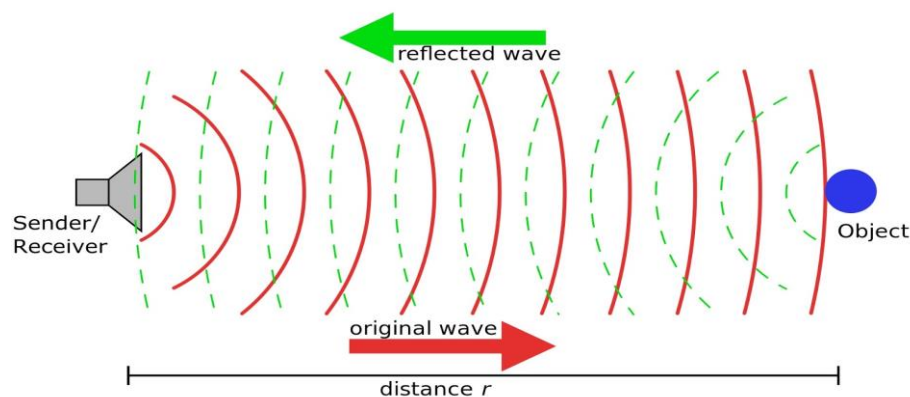


Figure 2. Ultrasonic Wave Propagation

- Effect of Temperature

The temperature plays a critical role in an accurate measurement of the distance of an object, while using ultrasonic sensor. The velocity of the sound wave propagation is obtained with respect to temperature by using the formula:

$$V = 331.5 + (0.607 * T) \text{ (m/s unit), where } T = \text{temperature in degree Celsius (}^{\circ}\text{C)}.$$

It is clear that the velocity (v) of the sound wave is directly dependent on the Temperature (T). As the temperature varies, the velocity of the sound waves also goes changing. Therefore, it is necessary to take an account of the temperature to get a reliable calculation of the distance between the sensor and the target object.

- Attenuation

The word attenuation is defined as the gradual loss of energy (intensity) of ultrasonic waves when it is propagated into a medium [Attenuation, 2015]. The intensity of ultrasonic waves decreases proportionally with the distance when it propagates into the air medium. This is caused by scattering, and absorption of the sound energy. Scattering is the reflection of sound wave other than its original direction and absorption is defined as the transformation of sound energy to other forms [Ultrasonic sensor, 2015]. The figure (3) shows that when the attenuation rate is bigger, the frequency of the ultrasonic wave is higher and the waves reach the shorter distance.

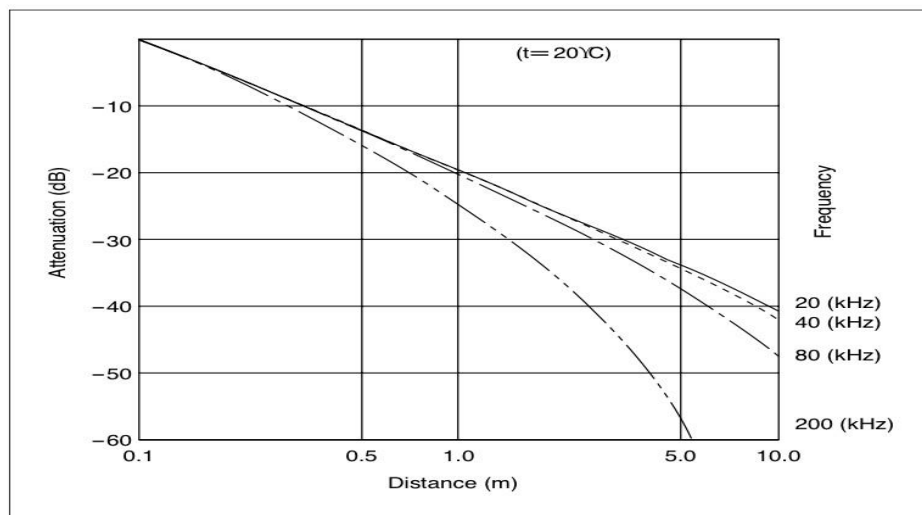


Figure 3. Attenuation characteristics of sound pressure with distance

2.2 Importance and drawbacks of ultrasonic sensors

Ultrasonic sensor has unique features over other sensors due to which it is applied in many application areas. It can easily detect and measure the distance of the moving object, which is impossible to get such a sensing function in other sensing devices. Thus, it is ideal for liquid level measurement or any other linear motion monitoring applications. Because of resistance to the external disturbance factors such as vibration, noise, EMI and Infrared radiation, it produces more accurate measurement. The sensing function of ultrasonic sensor is less affected by the surface and material of an object, and even not dependent on the object surface color or optical reflectivity of an object. For example, sensing of different colored glass plates such as a white plastic dish or a clear glass plate or a shiny aluminum plate remain the same.

Even though higher sensitivity detection of ultrasonic sensor offers several advantages, sometime its high sensitivity may turn to weakness, as result, it may yield false response when detecting even an innocuous objects such as passing insects or birds within the sensing field or even the hissing sound produced by the air hoses or relief valves. In comparing to other sensors, it has less sensing distance measurement; therefore, it is not possible to apply for long-range measurement. The ultrasonic sensors are based on the principle that emits the ultrasonic waves and reflects back to itself after detecting to an object. Hence, the response of ultrasonic sensor is directly affected by the change in the environmental factors such as temperature, air pressure, humidity, air turbulence, and even the surface of the objects. For example, the reflection of sound waves on the smooth surface such as glass is more efficient than the rough surface. And, some objects such as clothes absorb the ultrasonic waves instead of bounce back to the source. Thus, It is not applicable to an object having less reflectivity.

2.3 HC-SR04 Ultrasonic Module

Ultrasonic Module HC-SR04 is a sensor, which is used for distance measurement, and has the capacity of 2 cm to 400 cm range of non-contact distance measurement function. Author in [Ultrasonic ranging module HC-SR04, 2015] states that the accuracy range of Module HC - SR04 reaches to 3 mm, and consequently, it provides the high accuracy range of the measurement. The sensor comprises four distinctive

sorts of pins with different functions, which are: VCC, Trig, Echo, and GND. Furthermore, the Module HC- SR04 consists of three parts - ultrasonic transmitter, ultrasonic receiver, and control circuit. The ultrasonic transmitter emits the waves towards an object, and then receiver receives the reflected waves after reflecting back from the target object.



Figure 4. Ultrasonic Sensor: HC-SR04

2.3.1 Electric Parameters and Its Limitation:

Despite the fact that the module HC - SR04 is easy to apply and set up, it is important to know the limitation of the module and to adopt some safety measures before taken into practice. Otherwise, it may bring accident or may not work properly during the operation. The table 1 shows the electric parameter and its limitation.

Table 1. HC-SR04 Electric Parameters [Ultrasonic Ranging Module HC-SR04, 2015]

Electric Parameters	HC-SR04 Ultrasonic Module
Operating Voltage	DC 5V
Operating Current	15 mA
Operating Frequency	40 kHz
Maximum range	400cm
Minimum range	4 cm
Measuring Angle	15 degree
Input Trigger Signal	10 us TTL plus
Output Echo Signal	Input TTL level signal and the range in proportion
Dimension	45*25*15 mm

2.3.2 Operation Principle

At the point when the Trig pin of the module HC-SR04 receives a beat of high voltage (5V) for at least 10 us (microsecond), the sensor becomes active for ranging, therefore, the module sends out an 8 cycle burst of ultrasound waves at 40 kHz, and wait for the reflected ultrasonic burst [Ultrasonic Ranging Module HC-SR04, 2015]. When the sensor detects ultrasonic waves in the receiver, it sets the Echo pin to high (5V) and postponement for a period (width), which is directly dependent to the covered length by the wave. And, the range can be measured by applying the time interim between sending trigger signal and receiving echoes signal.

Time = Width of Echo pulse, (in microsecond)

Formula:

Distance = Time / 58.2 (in centimeter)

Distance = Time / 148 (in inches)

Range = (high level time * velocity (340 m/S)) / 2

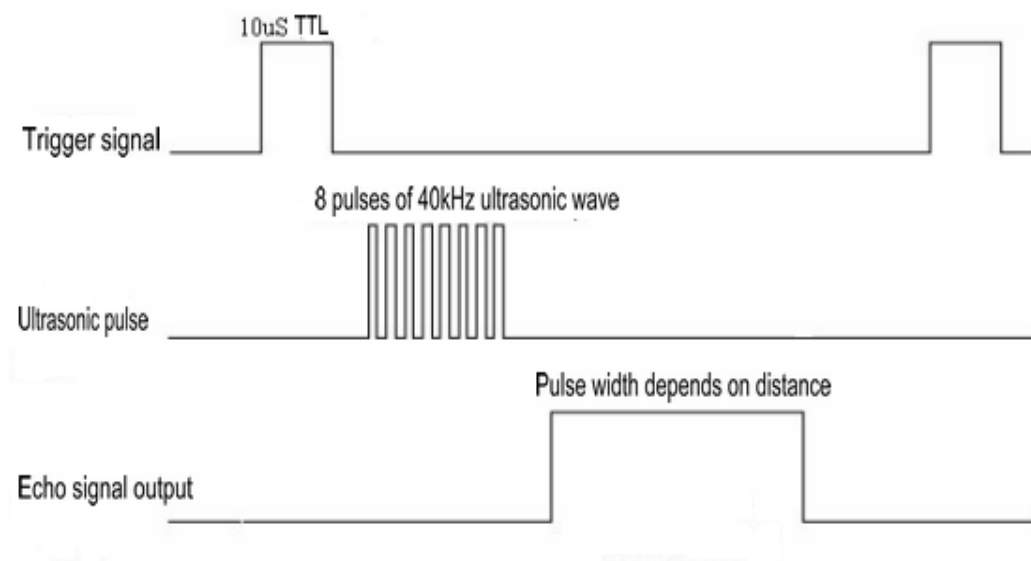


Figure 5. Ultrasonic HR-SR04 Module Timing Diagram [Ultrasonic Ranging Module HC-SR04, 2015]

2.3.3 Practical Consideration

To get an accurate measurement, the practical consideration should be taken into account during the sensing operation. One of the importance methodologies that ought be considered during the distance measurement of an object is described in brief as below.

- Object Positioning

The ultrasonic ranging module HC -SR04 cannot measure the distance of an object accurately if the object: a) is 400 cm far away from the sensor, b) has the reflective surface at the shallow angle (less than 45 degree), therefore, the ultrasonic waves cannot return back to sensor, and c) is too tiny to reflect back ultrasonic waves to the sensor. Hence, these things must be avoided to obtain accurate results.

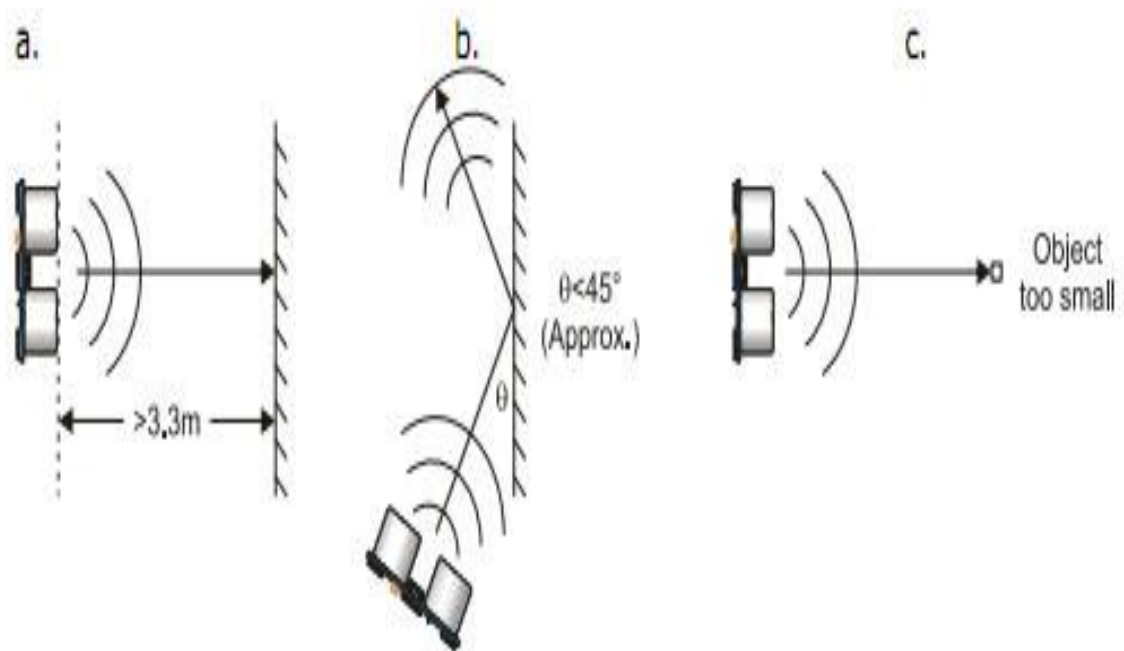


Figure 6. Error Case in Object Position

3 ARDUINO UNO

It is an open source microcontroller board, which is based on the ATmega328 chips. It is been designed for developing a numerous electronic devices as well as interactive prototype that can be used for controlling and monitoring the physical elements such as moisture, temperature, motion, the level of liquid, and much more. It is a small, portable, and cheap device which comprises 14 digital input/output pins (out of which 6 can be utilized as pulse-width modulation outputs - PWM), 6 analog inputs, a crystal oscillator of 16 MHz, a USB cable connection, an external power jack, an ICSP programming header, and a reset button [Arduino Uno, 2015]. It can be buy from the Arduino official website or any other electronic products distributors such as Element14, RS Component, and so on. To get the Arduino started, simply need to supply the power by using USB cable from a computer or just need to power it with AC-to- DC adaptor directly from the power source. Unlike from the previous version of the Arduino, it consists of Atmega8U2 programmed instead of FTDI as a USB-to- serial converter.

As an open source, for any hardware enthusiast, the references for hardware designs of an Arduino are freely available on the official website (<http://arduino.cc/>) [Arduino Uno, 2015]. The Arduino IDE, which is written in java programming language, is a multi-platform application. Arduino IDE is designed to program the Arduino board as per the need of the user projects. The Arduino IDE is come up with the great software library, which features much more easily way to the users, while writing the codes for the common input and output operations. To make an Arduino as program runnable (real-time operation), it requires just two simple functions:

- 1) setup (): starts once to setup the parameters, such as communication rate and pin mode.
- 2) loop (): run continuously unless the arduino power off.

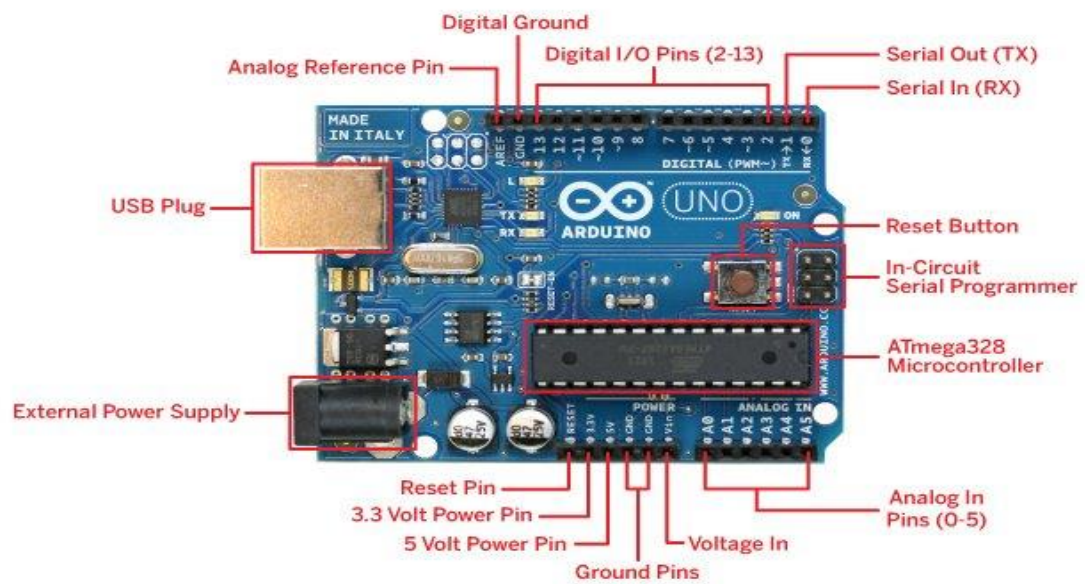


Figure 7. Arduino Uno and its Components

3.1 Arduino Technical Specification

Even though Arduino is widely used for many Internet of Things (IoT) purposes, it has some limitations, which is very important to figure out before taken in to the configuration. The technical specification of Arduino Uno board is listed in the table 3.

Table 2: Technical Specification of Arduino Uno. [Arduino Uno, 2015]

Microcontroller	ATmega328
Working Voltage	5 V
Input Voltage (recommended)	7 V - 12 V

Input Voltage (limitation)	6 V - 20 V
Digital I/O Pins	14 (out of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB, 0.5 KB used by boot loader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

3.2 Input and Output Operation

Arduino Uno consists of 14 digital pins. Each of them is responsible for Input or Output functions. All pins, normally, operate at 5 voltage and each input pin receives the maximum current of 40 mA and the same range of current is produced by the output pins. The functions of each input and output ports are described in brief as below:

- Serial Out (TX)/In (RX): They are used for communicating the Arduino and other devices such as computer, laptop, and so on. Serial Output port (TX) is used to transmit the serial data whereas Serial Input port (RX) is used for receiving the serial data. The syntax for these ports is: `if (serial) {...}`.
- Pin No. (2 & 3): These pins are responsible for external interruption. The user input data can be easily monitored, and read the rotary code by applying the `attachInterrupt()` method.
- Pin 4, 5, 6, 9, 10, and 11: They give 8-bit pulse width modulation output using the `analogWrite(pin no., value)` function.
- 10 (SS) pin, 11 (MOSI) pin, 12 (MISO) pin, 13 (SCK) pin: These ports are known as SPI ports, which are responsible for communicating between peripheral devices. Pin 12 (MISO) sends data to master device (microcontroller). Pin 11 (MOSI) is a master line for sending data to peripheral devices. Pin 13 (SCK) synchronizes the data transmission. Pin 10 (SS) is used by master device to enable and disable the devices.
- Pin 13: It is LED pin for the Arduino Uno. The pin gets ON when the LED value becomes high and gets OFF when LED value falls to low.
- AREF: It is the reference voltage for analog input. The syntax is: `analogReference(type)`.
- Reset: This pin is used to reset the program to Arduino Uno.

3.3 Power Supply Consideration

Arduino Uno can be supplied with the power directly either from the external power source using an AC - to - DC adaptor or through the USB cable connection. Although the Uno can work with an external voltage between 6 V to 20 V, 7 V to 12 V is the recommended supply voltage. It may burst or damage the hardware if the supplying power is more than 12 V because the voltage regulator becomes overheated in excess of 12 V. On other hand, it may not be stable, and may not work properly, if the Uno is provided with less than 7V because the 5V power pin may generate lower than 5 V. Therefore, it is necessary to know about the power supply to Arduino before taking it into practice. Arduino consists of following power pins (Arduino Uno, 2015):

VIN pin: It is an input voltage to an Arduino when power is provided directly from external power jack rather than USB cable connection. The VIN power pin remains unregulated unless the Arduino is supplied with the power from the external power jack.

5V pin: This power pin produces an output of 5V from regulator board in which the power is supplied either of 3 power ports: 1) an external DC power supply (7 - 12 V), 2) USB cable connector (5V), or 3) VIN power pin (7 -12 V). It can also be used to power the external components of the 5V connection.

3V3 pin: It is an output of the on-board 3.3V regulator, which takes in the current of maximum 50 mA. It can be used to supply power for external component that need 3.3V.

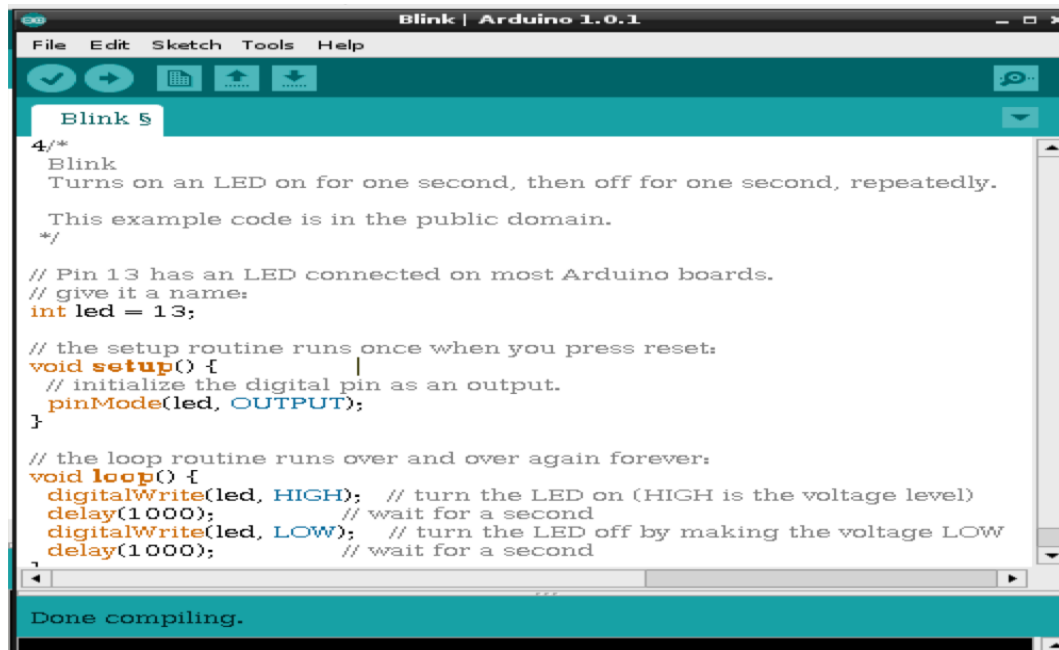
GND: It is a ground pin to the Arduino for safety functions.

IOREF: It is a voltage reference to the Input/output operation of the Arduino board. The IOREF pin receives the voltage from the Arduino board.

3.4 Arduino Software

It as an open-source software, therefore, the software can be freely downloaded from the Arduino official website. The Arduino software IDE is written in Java, which is easy

to write codes with and upload the programs to the Arduino. The Arduino IDE is a cross-platform application software. Hence, it can run in many operating systems such as Linux, Mac OS X, and Windows. The stable version 1.6.1 is available and is compatible with all versions of Arduino boards. Unlike other IDEs, the Arduino IDE has the code editor features such as brace matching (helps to navigate through the codes and point out the improper matching in the codes), syntax highlighting (displays source code in different colors and fonts based on category of term), and automatic indentation and also has the capacity of compiling and uploading the programming codes to the device. The IDE consists of a software library that makes it more convenient for the user to write the codes for input and output operations. The programs on the Arduino are coded in C++ language and a written code or program is known as a sketch.



```

4./*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}

```

Done compiling.

Figure 8. Arduino IDE showing 'blink' program

4 RASPBERRY PI

Raspberry Pi is a cheap, and a small single board computer developed in the UK by the Raspberry Pi Foundation with the aim of stimulating the basic computer science for the school children. This credit-card sized computer is capable of many functions as the desktop PC does, such as games, word processing, and spreadsheet and it can also play high definition video as a normal computer. It can run Linux based operating systems such as Raspbian, Ubuntu Snappy core, Puppy Linux, and others. It is built up with the powerful processor- Broadcom BCM2835 System-on-Chip that consists of ARM1176JZFS with floating point, running at 700MHz, and a Video core 4 GPU [Raspberry Pi, Element14]. The GPU is capable of 1Gpixel/s, 1.5Gtexel/s or 24 GFLOPs of general-purpose compute performance, and provides the Blue Ray quality video using H.264 at 40MBits/S.

The Raspberry Pi does not have a built-in data storage device such as hard disk, or hard drive. Thus, an external SD card with a minimum 4 GB size is needed to run the Raspberry Pi. It can support up to 32 GB card. It has 516 MB RAM with a 700 MHz processor. The two built-in USB ports are used for power supply to other devices like Arduino and also to connect the mouse, and the keyboard. It also has one 10/100Mbps Ethernet port for Internet connection. Apart from these, it consists of 26 General Purpose Input/ Output (GPIO) pins (17 out of them are responsible for input or output functions, and the rest are power and ground pins. In this project, the author used Raspberry Pi Model B.

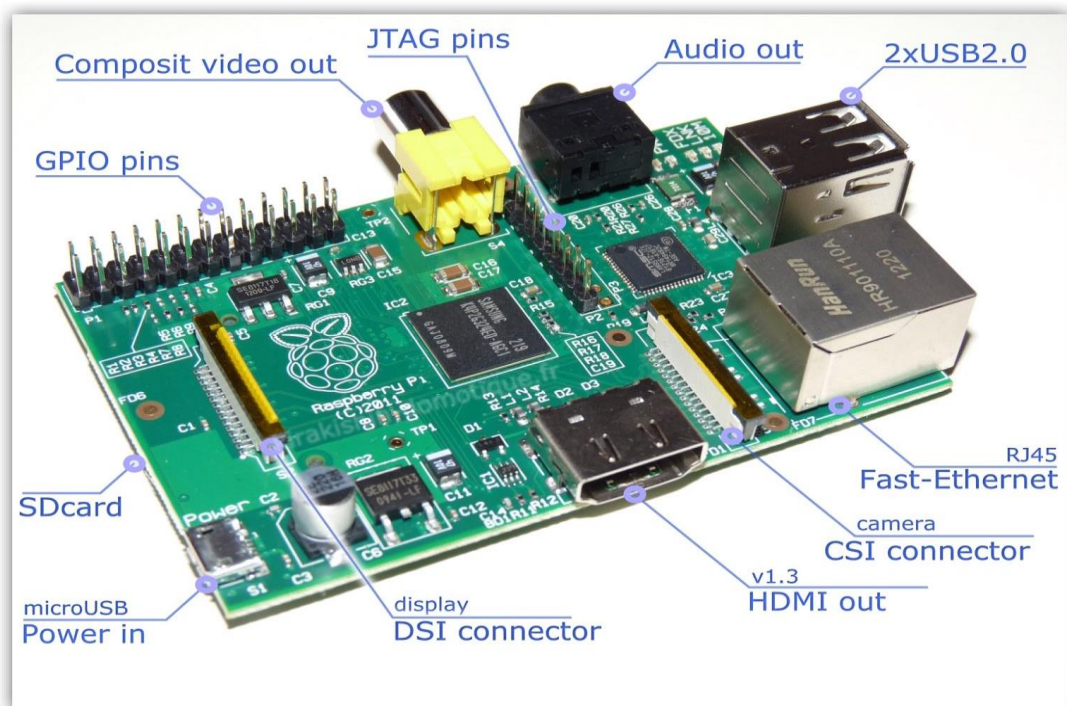


Figure 9. Raspberry Pi Model B

4.1 Modular Structure and Specification

Although Raspberry Pi is small like a credit card, it can perform the tasks as a desktop computer does. Table 3 demonstrates the hardware specification comparison between Raspberry Pi Model A and Mode B.

Table 3. Raspberry Pi Model A and B specification [Raspberry Pi, Element 14 2015]

Specification	Model A	Model B
System on Chip (SoC)	Broadcom BCM2835	Broadcom BCM2835
Central Processing Unit (CPU)	700 MHz ARM1176JZF-S core	700 MHz ARM1176JZF-S core
Memory (SDRAM)	256 MB	512 MB
No. Of USB port	1	2
Ethernet Port	None	One 10/100 Ethernet (RJ45)
Power Rating	300 mA	700mA
Storage Device	SD/MMC/SDIO card slot	SD/MMC/SDIO card slot
Price	25 US dollar	35 US dollar
Weight	45 g	45 g

4.2 Power Supply Consideration

, Raspberry Pi can be supplied with the electric power in three different ways. Firstly, the power to Raspberry Pi is given by using a micro USB port as a power connector, which runs on 5V power. Secondly, power can be supplied through the USB hub or

GPIO header Pin 2. The supplied voltage from the GPIO pin 2 is 5 V and, finally, from a single USB hub which is around 3.3 V and may not provide enough power to run the Pi. Powering the Pi, therefore, via the micro USB port is a good practice. Usually, 700mA is needed to operate smoothly to the Raspberry Pi Model B. However, the power requirement sometimes increases when all the interfaces are brought in use. Power rating for each component differs from the others, for example, the HDMI port requires 50 mA, the Camera Module needs 250 mA, the keyboard and the mouse need 100 mA or over, and 50 mA is distributed across the GPIO header. Some consideration during the power supply include:

- It is not a good practice to power the Pi from two different power sources (USB hub and external power adapter) simultaneously as this may seriously damage the Pi
- Raspberry Pi recognizes some USB hubs as ‘back feeds’. “Backfeeds” may cause malfunction to the computer, therefore, it is very important to be aware of the “back feeds”.

4.3 GPIO Hardware Connector

The Model B includes 26 GPIO pins, which feature the interaction between the Pi and the outside world. As per need, these pins can be programmed and used for either input or output purposes. Seventeen pins, among the total GPIO pins, are used for input functions (to input the sensor data or signal from other devices) and output functions (to turn on LED or send data to other devices). Pin 1 and 2 are mainly used for power supply, which provide 5V and 3.3V respectively. Figure 6 shows the functions of each GPIO pins in the Model B Raspberry Pi.

GPIO#	2nd fun	pin#		pin#	2nd fun	GPIO#
-	+3V3	1		2	+5V	-
GPIO2	SDA1 (I2C)	3		4	+5V	-
GPIO3	SCL1 (I2C)	5		6	GND	-
GPIO4	GCLK	7		8	TXD0 (UART)	GPIO14
-	GND	9		10	RXD0 (UART)	GPIO15
GPIO17	GEN0	11		12	GEN1	GPIO18
GPIO27	GEN2	13		14	GND	-
GPIO22	GEN3	15		16	GEN4	GPIO23
-	+3V3	17		18	GEN5	GPIO24
GPIO10	MOSI (SPI)	19		20	GND	-
GPIO9	MISO (SPI)	21		22	GEN6	GPIO25
GPIO11	SCLK (SPI)	23		24	CE0_N (SPI)	GPIO8
-	GND	25		26	CE1_N (SPI)	GPIO7

Figure 10. GPIO Header in Model B Pi [Raspberry Pi, Wikipedia 2015]

5 SOFTWARE TOOLS OVERVIEW

This section describes the different software tools that have been used for developing the project are described.

5.1 Raspbian

Raspbian is a Debian-based free operating system, specially designed for the hardware of the micro-computer, Raspberry Pi. The operating system is a collection of programs and utilities that manages the computer hardware and software components and provides the services to operate the computer [Operating System – Computer Hope, 2015]. For Raspberry Pi, a wide range of choice of OS is available such as Pidora, RISC OS, Arch Linux ARM and Raspbian. Among them, Raspbian has become one of the best choices because it is easy to setup, easy to use, and has great performance. It comes up with lot of built-in software tools and apps. For example, Sonic Pi software application that helps to learn the coding fundamentals. Raspbian is compatible with the Pi peripheral devices, such Camera module, GPIO header, Audio and Video ports, and so on. Apart from that, it also supports the recommended overclocking (push the original speed of CPU to higher speed). Raspbian was created by a small team of dedicated software developers, but is not associated with the Raspberry Pi foundation [Raspbian – raspbian.org, 2015].

5.2 Python

Python is the most widely used programming language that can be used for general-purpose as well as high-level programming. This general-purpose programming language is defined as the computer language that is able to write software in a wide variety of application domains, while high-level language refers to any computer programming languages that are human readable and independent of the architecture of computer hardware. Because of its clean and straightforward syntax, the language is easy to use and write application programs. Unlike other programming languages, it supports several programming paradigms such as object-oriented programming, imperative and functional programming as well as procedural programming. Programming paradigm is the basic style or design for making the structure and elements of computer programs. Python code is compiled by using the Python interpreter or third party software applications such as Py2exe or Pyinstaller.

5.3 MySQL server

The MySQL server is the most popular open source relational database management system, which was developed by Oracle Corporation. Being open source, it can be downloaded freely and can be modified as per the one's needs. The full form of 'MySQL' is My Structured Query Language. MySQL [DB-Engines Ranking of Relational DBMS, May 2015] is the world's second widely used for relational database system. In the world of database management system, it has become the most popular choice for the web applications and the open source operating system, especially, Linux, Debian, FreeBSD, and Raspbian. In addition to being an open source operating system, it is compatible with many other OS such as Windows and Mac OS X. It can handle a large number of data about 50 million rows (or more than it) and has the default file size limit up to 4GB but can be increased up to 8 TB, if needed. For this project, MySQL version 5.5 was installed on the Raspberry Pi.

5.4 VNC Server

The VNC (Virtual Networking Computing) server allows to remotely control one computer (server) from another computer (user's computer). VNC uses Remote Frame Buffer (RFB) protocols for graphical desktop sharing. A connection between server and client requires a VNC client tool, for example, a VNC viewer on the user's computer to control and view the remote computer (VNC server). In this project, 'tightvncserver' is used for the remote connection to the server, but there is no encryption connection within this free version. Thus, in the real life applications the VNC server with encryption is recommended for security reasons. The server 'tightvncserver' is used to share graphical desktop, modify and check the database system on the remote computer (Raspberry Pi).

5.5 IIS Express

IIS Express is a web server developed by Microsoft, which is basically running on the Windows operating system. It is responsible for the client requests and supports various protocols such as HTTP, HTTPS, FTP, FTPS, SMTP, and NNPT. IIS Express is the light version of the IIS server, which is easy to install and does not require any administrative account to run and debug web applications from the Visual Studio. It can be downloaded side by side with the Visual Studio and ASP .NET development tools. In this project, IIS Express along with Microsoft Visual Studio was installed for building the web page. To run and test the web application (web page) that is created by using ASP .NET, and C#, the IIS Express server was used in the project.

5.6 Visual Studio

Visual Studio is an Integrated Development Environment (IDE) created by Microsoft, and is used to build the web applications, web services and website, primarily running on the Windows operating system. Unlike other IDEs, it includes features such as debugger, code editor (supports syntax highlighting and code completion), Windows form designer (build GUI applications by using a window form) and other essential tools such as Team Explorer, Solution Explorer, Server Explorer, etc. However, it is only compatible with the Window OS. Many programming languages such as Visual basic, C#, C++, C, and so on can run /debug on the Visual Studio. In this project, Visual Studio Express 2013 was installed to develop the web page.

5.7 HTML

HTML is the collection of markup symbols and codes which are used to develop web pages. HTML stands for HyperText Markup Language. It describes the structure and design of web page by using variety of attributes and tags. Every web site is built up with a combination of different HTML elements. Each element consists of opening and closing tags. For example:

```

<!DOCTYPE html>

<html>

<body>

<h1> This is Header</h1>

<p>This is Paragraph.</p>

</body>

</html>

```

See the source HTML code for this project in Appendix 3.

5.8 CSS

Cascading Style Sheets (CSS) are one of the most widely used markup languages for creating beautiful and attractive web pages. CSS helps to make the layout of the web pages and to define text style, text positioning, text spacing, font color, and table size as well to format other aspects of the webpage. For example, CSS can be applied to define the cell padding of the table cells, as well as the color and the style of a table's border. Instead of defining CSS in each line of HTML pages, CSS can be defined only in the CSS document (folder) and can be used for several HTML pages at a once which really makes it easy to use for the web developer. CSS used in this project can be seen in Appendix 4.

5.9 ASP.NET

ASP.NET is the server-side web framework, which is designed and developed by Microsoft for building the web applications, especially dynamic web pages, web applications, and web services. The ASP .NET provides mainly four types of development frameworks: 1) ASP .NET Web Form 2) ASP .NET MVC, 3) ASP .NET Web Page, and 4) ASP .NET Single Page Application. Using any of these four frameworks, the web applications can be built as per the one's need. Apart from these frameworks, it also offers some other extension frameworks such as ASP .NET Web API (HTTP API for web services), ASP .NET SingleR provides a library for real time

communication between server and client and so on. To display the current volume of liquid on the web page, the 'ASP .NET Web Page' framework was used in the project. Finally, a great built-in feature of the ASP .NET framework, the Grid View was applied to show the data in the table. The source code of ASP .NET file can be seen in Appendix 7.

5.10 C Sharp (C#)

C Sharp (C#) is an object-oriented, high-level programming language, which is designed and developed by Microsoft. When the word object-oriented is attached to a programming language, the language should include three main features such as Encapsulation, Inheritance, and Polymorphism. Thus, like other object-oriented programming language, C# supports all of the above object-oriented concepts. Encapsulation is defined as the process of bringing one or more items within a physical or logical package. Similarly, inheritance means that one class (derived class) can be derived from another class (base class), which provides the chance to reuse the code functionality and increase the implementation time. Therefore, it is easy to create beautiful software applications. In programming languages, polymorphism refers to the capacity to provide a single interface for different object types. C# is also the hybrid of C, and C++ languages and has some similar features to the Java programming language. C# is designed to build diverse software applications such as web applications, mobile applications, websites and games by using Microsoft .NET frameworks. In the project, C# was used to display the data in the web page (See Appendix 5).

6 PROJECT DESIGN AND DEVELOPMENT

The main idea of this project was to track and monitor the level of liquid in a tank remotely. The work aimed to measure the current volume of liquid in the tank and send the data to the central monitoring station, using different hardware devices and software tools. Apart from this, the project aimed to develop a web page designed to display the data based on the location. By using a standard Internet browser, the authorized person can retrieve the information from anywhere in the world.

6.1 Project System Design

The proposed system consists of the Central Monitoring Station and Remote Network Site Topology. In the remote network site topology, the different hardware components are equipped to obtain the desired results, which are described step-by-step as below.

- **HC-SR04 Ultrasonic sensor:** Because it is easy to setup, and also for economic reasons, in the prototype, the author used the HC-SR04 Ultrasonic sensor to measure the current volume of liquid. However, the sensor can be replaced with the other sensors depending on the physical factors such as tank shape and size, and type of liquid.
- **Arduino Uno (Microcontroller):** It acts as a microcontroller, which collects the information from the HC-SR04 ultrasonic sensor and forwards them to the database server - Raspberry Pi.
- **Raspberry Pi (Database server):** In the Raspberry Pi, a MySQL server is installed for storing the liquid level sensing information. MySQL also includes the VNC server for remote graphical sharing. The Raspberry Pi is connected with either Wi-Fi Network or Cellular network for communicating with to Central Monitoring Station. Raspberry Pi should have a static IP address to connect to the Central Station.

The Central Monitoring Station is comprised of the personal computer with data monitoring software tools such as MySQL Workbench and web services including a

web page to display the liquid level data. The central Monitoring Station connects to the remote servers, Raspberry Pi in different locations, and monitors, stores, analyzes and backs up the sensing data and finally provides the data to the authorized user.

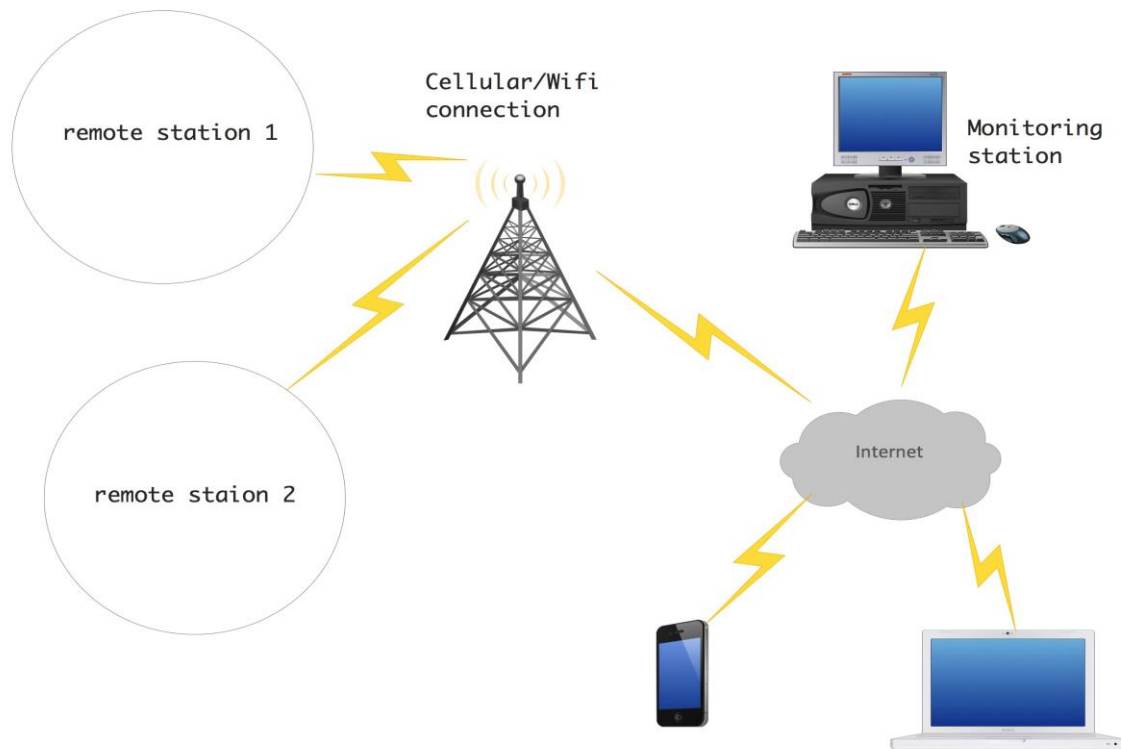


Figure 11. Project System design

6.2 Remote Site Network Development

To develop the remote area network topology, some fundamental resources and hardware components were used, which are discussed below.

- Internet: Since the whole idea of the project is to fetch the data from a remote server to a central monitoring station, regular Internet connection with a minimum speed to run the database server is required to run the project. Uninterrupted Internet connection can be used through Wi-Fi connection or cellular network.

- Power source: To run the electronic devices such as sensor, Arduino, and Raspberry Pi, a regular electric power source should be available in the remote area. Figure 12 shows the remote site topology:

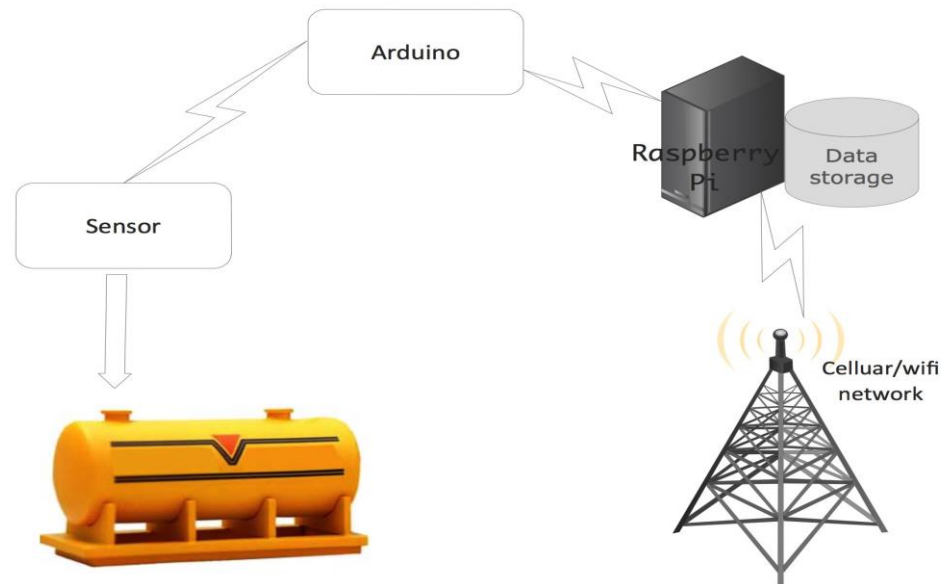


Figure 12. Remote Site Network Topology

6.2.1 Arduino Modification with HC-SR04 Sensor

In the section, the connection between Arduino and the HC-SR04 sensor is described. The used devices and software applications for the Arduino modification with the sensor are:

- Male to female jumper wires
- HC-SR04 ultrasonic sensor
- Arduino Uno (and Arduino IDE)
- Vertical cylindrical vessel (taken as glass)
- Breadboard (optional)

The HC-SR04 ultrasonic module consists of four pins: GND pin, VCC pin, Trig pin, and Echo pin, each with specified functions. By using the male to female jumper wire, the

GND and VCC pin of the module are connected to GND pin and the 5 V power supply pin of the Arduino board respectively. Likewise, the Trig pin is defined for the output signal, and the echo pin is defined for input signal, and these are connected to digital pins 8 and 7 correspondingly. Figure 13 shows the connection between the HC-SR04 sensor and the Arduino board.

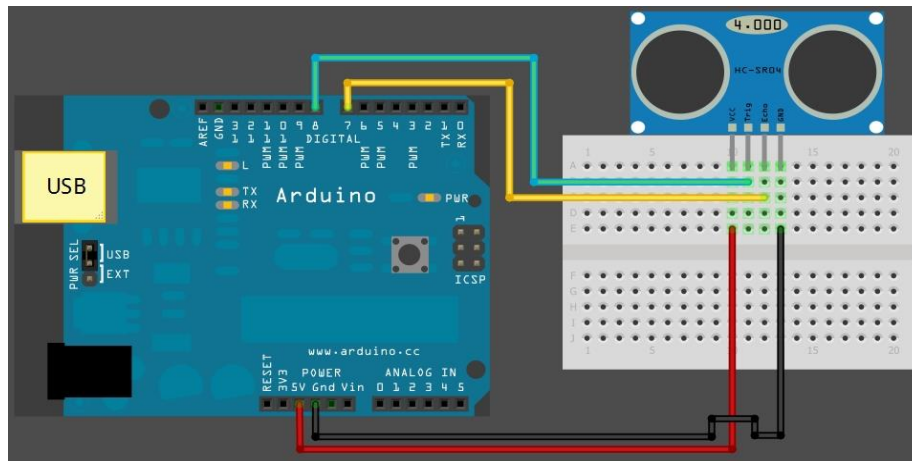


Figure 13. Connection between HC-SR04 and Arduino Uno board

In this project, Arduino is used as a microcontroller, which converts physical data such as volume, distance of the tank into the machine-readable data (digital data). Therefore, the Arduino is programmed to find the current volume of liquid in a tank, and then forward these data to the database system. To program the Arduino board, the Arduino IDE was installed in the Raspberry Pi and was coded with the C++ language to obtain data about the current volume of liquid in the tank. The code used to modify the Arduino to find the current volume of liquid in the vertical cylindrical vessel is listed in Appendix 1. More details about the vessel shapes and its implementation are discussed later in this chapter.

As the code is compiled and uploaded to the Arduino board, it reads the data (level of liquid) from the vessel using the serial monitor. The serial monitor is the communication channel between an Arduino to any other electronic devices, for example, Raspberry Pi, laptop and much more. The serial port `-/dev/ttyACMO` is used as communication path between Raspberry Pi and Arduino Uno. Figure 14 demonstrates the output of the Arduino serial monitor.



Figure 14. Screenshot of Arduino Serial Monitor Output

6.2.2 Raspberry Pi Modification

In the project, the Raspberry Pi was used as a remote server, especially for database and VNC server. Raspbian was used as an operating system for the Raspberry Pi. The software was downloaded from the Raspberry Pi's official website – (<https://www.raspberrypi.org/downloads/>). Before installing the database and the VNC server, the Raspberry Pi was connected with the Internet and was configured with the static IP address: **192.168.11.7**. A free mobile application, 'Fing' was used to find the dynamic IP address of the Raspberry Pi. By using SSH connection to the Raspberry Pi, Author used the following steps to make the static IP address of Raspberry Pi.

pi@raspberrypi ~ \$ sudo nano /etc/network/interfaces

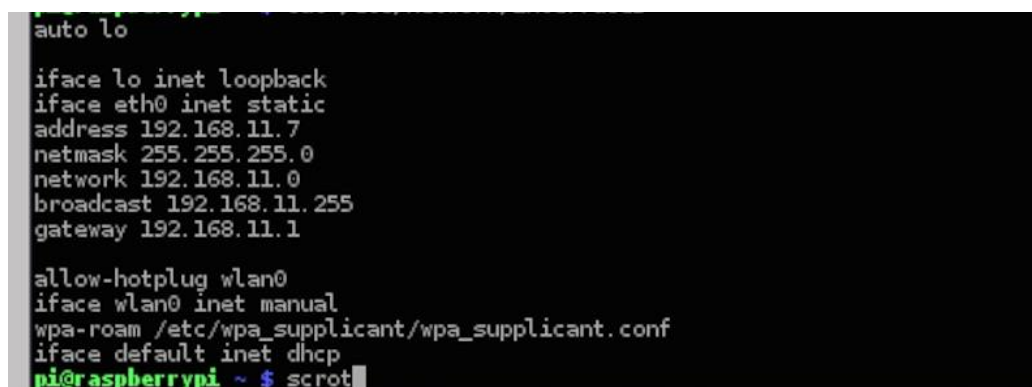


Figure 15. Static IP address Configuration

6.2.2.1 VNC Server Configuration

The light version of the VNC server - 'tightvncserver' was installed (recommended to use the higher version for the better security reason) and run the server using the following commands.

pi@raspberrypi ~ \$ sudo apt-get install tightvncserver (install server), and

pi@raspberrypi ~ \$ tightvncserver (start to run the server on the Pi).

Now, the Raspberry Pi is ready to connect anywhere around the world with the any device having VNC client such as VNC viewer. For a secure connection, it requires a username or IP address, and VNC password of the server. Figure 16 shows the desktop monitoring of VNC server (Raspberry Pi) from the central station's computer.

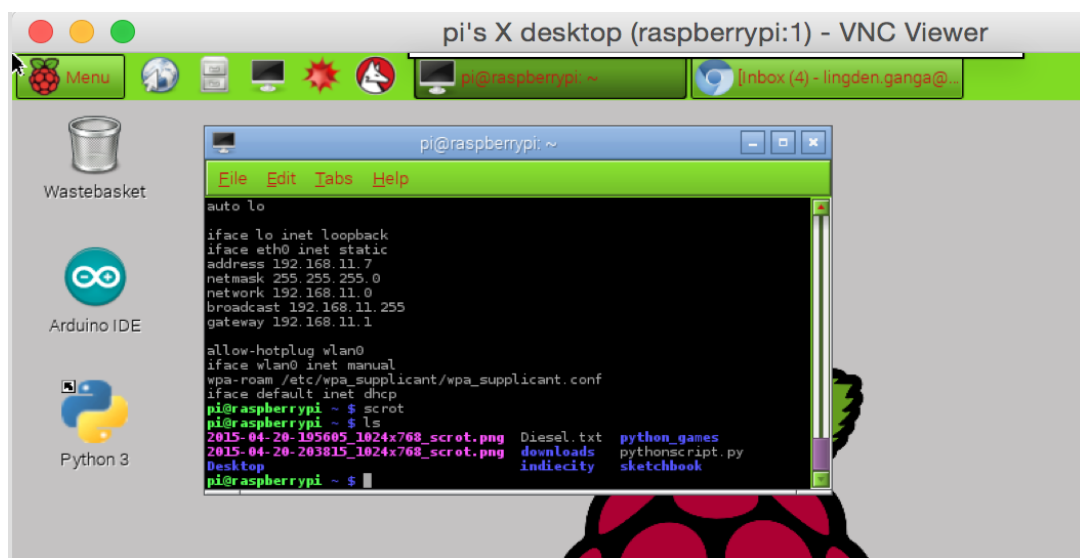


Figure 16. Raspberry Pi Desktop in VNC viewer

6.2.2.2 Database Server Configuration

As the main objective of the project was to measure levels of liquid in the tank, collecting the sensing data in the database was the very important part of the project. Thus, to perform the desired task, the database design, and server selection was a crucial steps in the development of the project. The open source database server,

MySQL server, which is compatible with the used Raspberry operating system - Raspbian was installed by using the below command.

pi@raspberrypi ~ \$ sudo apt-get install mysql-server --fix-missing

To store the sensing data (level of liquid) into the database, a database named -Turku was created on the server on system privilege mode for the user. Inside the Turku database, a 'Diesel' table with date, time, volume, and location columns was created, but the data in the table was inserted manually (by executing the Python code each time) at this point, which is shown in Figure 17.

```
mysql> select * from Diesel;
+----+-----+-----+-----+-----+
| Id | Date       | Time       | Volume_In_Litre | location |
+----+-----+-----+-----+-----+
| 1  | 2015-04-22 | 13:28:12   | 0.42            | Turku    |
| 2  | 2015-04-22 | 13:30:58   | 0.05            | Turku    |
| 3  | 2015-04-22 | 13:33:24   | 0.05            | Turku    |
| 4  | 2015-04-22 | 13:35:57   | 0.11            | Turku    |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

Figure 17. Manually inserted data into database

6.3 Arduino and Raspberry Pi Communication

There are many ways to connect Arduino and Raspberry Pi, such as the GPIO and Serial Pin connector, the USB cable connection, and the I2C connection. Among these connection types, in the project, the author used the USB cable connection between Arduino and Raspberry Pi because it does not need external hardware; the USB cable is enough to establish this communication. Between these two devices, Arduino and Raspberry Pi, two ways of communication are possible, i.e., from Arduino to Raspberry Pi and vice versa. To allow data reading from Arduino to Raspberry Pi, pySerial (python library for serial communication) was installed in the Raspberry Pi. The pySerial allows to read and write to the serial port by using the Python programming language. The pySerial was installed with the command:

pi@raspberrypi ~ \$ sudo apt-get install python-serial

And, to retrieve the data from arduino, Raspberry Pi was programmed with the following code.

```
#!/usr/bin/python

import serial

device = serial.Serial('/dev/ttyACM0', 9600)

data = device.readline()
```

Furthermore, in order to store data in the MySQL server, the python-mysqldb library was also installed on the Raspberry Pi. Again, it was programmed with Python to insert the data that are collected by the Arduino. The Python script, which is used to fetch the data from Arduino to a database server, is available in Appendix 2. The crontab is also utilized to send the data to the server automatically on desire time- based. Crontab is the feature in a Linux-operating system that runs the code (the script) on a time basis. For example, in the project, the data is sent automatically to the central station in every 30 minutes using the following crontab:

***/30 * * * * python /home/pi/project.py**

6.4 Case Study based on the Liquid Tank Structure

While working with the HC-SR04 ultrasonic sensor as a tool for measuring the level of liquid in the tank, the different measuring methods should be taken into consideration because of tank shape variations. One practice for calculating the current volume of liquid of the tank may not be suitable for other tanks of different shapes. In this section, therefore, author has figured out how to find the level of liquid in different tank shapes using the HC-SR04 ultrasonic sensor.

- Vertical Horizontal Cylindrical Tank:

In the project, the Vertical Horizontal Cylindrical shape vessel is used as a tank for volume (level of liquid) measuring purpose. To obtain an accurate measurement of liquid level, the HC-SR04 sensor is put at the top of the tank vertically, so that the two

eyes of sensor face the bottom of the tank. The formula used to calculate the current liquid level in the tank type is demonstrated as below.

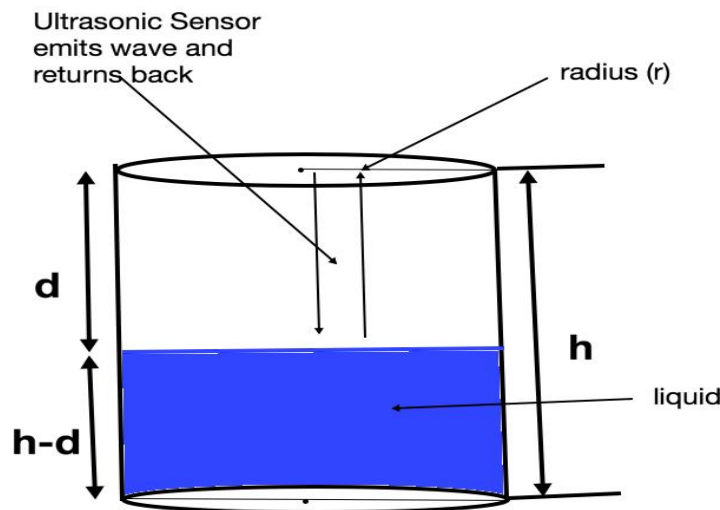


Figure 18. Liquid level measuring in Vertical Horizontal Cylindrical tank

Sensing distance = d

Height of the tank = h

Liquid level = $h-d$

Radius of the tank = r

Volume of the tank = $\pi r^2 h$

Volume of the liquid (current liquid level) = $\pi r^2 (h - d)$

- Horizontal Cylindrical Tank

In the liquid industries, several shapes of tanks are used to store liquids. Among them, the horizontal cylindrical tank shape is one of the most frequently used tank shape. In order to find the current liquid inside this type of the tank, the author has proposed the following procedure:

Diameter = d

Liquid level = $d - sd = h$

Radius = $d/2 = r$

Volume of liquid level = $[r^2 \cos^{-1}((r - h)/r)] - [(r - h)\sqrt{2rh - h^2}]$

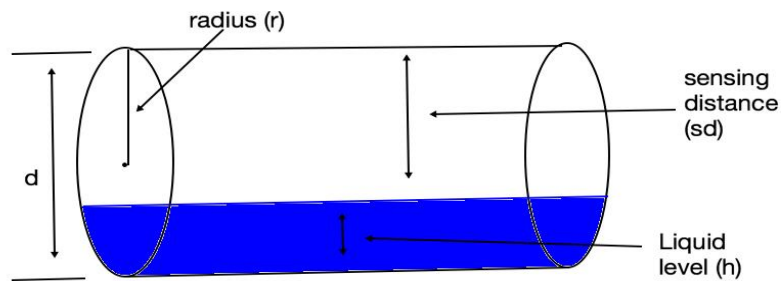


Figure 19. Liquid level measuring in Horizontal Cylindrical Tank

- Rectangle Tank

Using the following formula, the current volume of liquid is calculated in this tank.

Length = l

Breadth = b

Height = h

Sensing distance = d

Liquid volume = $l * b * (h - d)$

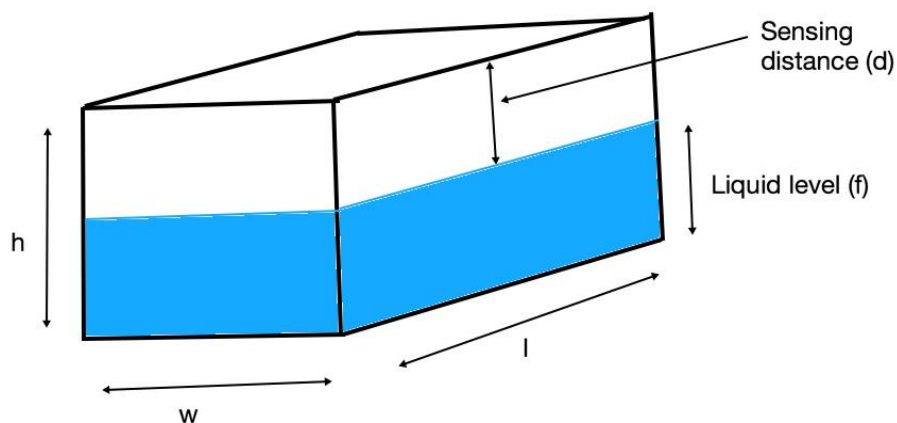


Figure 20. Liquid level measuring in Rectangle Tank

- Cube-shaped Tank

For the cube-shaped tank, the author has proposed with the following formula to obtain the current value of liquid inside the tank.

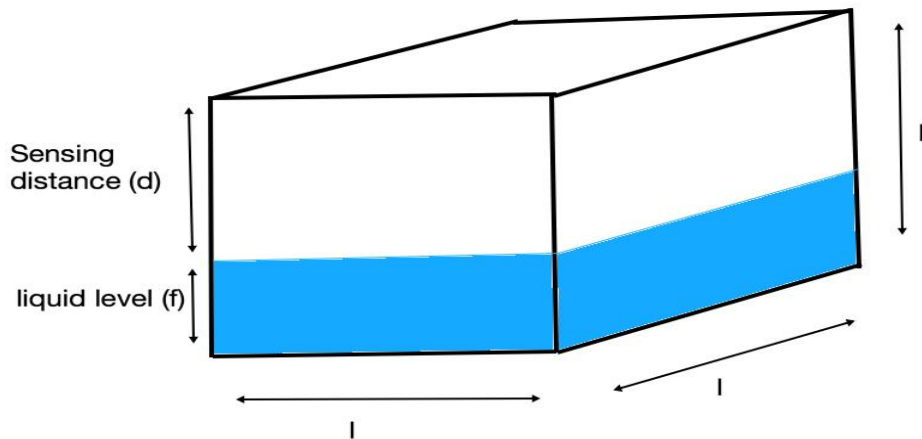


Figure 21. Liquid level measuring in cube tank

Length = l

Breadth = l

Height = l

Sensing distance = d

Volume of liquid = $l * l * (l - d)$

- Horizontal Oval Tank

It is one of the most frequently used tanks for liquid storing purposes. It is used in vehicles, for example liquid delivery vehicles. Thus, the following method is applied to this type of vessel, in order to find the current liquid level.

Diameter = h

Sensing distance = sd

Side length = x

Volume of tank = area of stadium shape * length = $(\pi^2 + d * x)l$

So, volume of liquid = $(\pi^2 + (d - sd) * x)l$

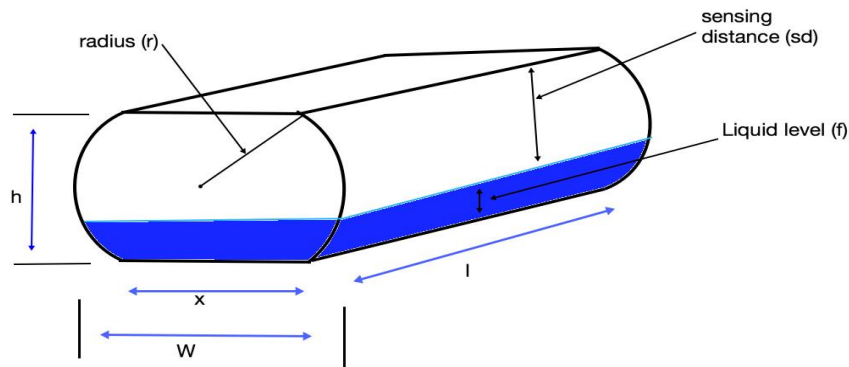


Figure 22. Liquid level measuring in Horizontal Oval Tank

6.5 Central Monitoring Station

The central monitoring station is built up with a PC (used for administrative purpose only), which consists of several remote monitoring software applications such as MySQL Workbench, SSH client and VNC client as well as a web service to display data. It is designed and developed to track the liquid level of different tanks from different locations. For example, suppose that there are five gas stations (or petrol pumps) in different five locations. In order to track the level of liquid from these five locations, the central monitoring station is designed to bring data from all the locations. The software tools used are described in detail as follows:

6.5.1 Database Connection

In the project, MySQL Workbench was used to manage the remote database system. It is the visual tool for Database Administrator (DBAs), and Database designers or architects for controlling, designing, and developing the database system. For the server configuration, database backup and recovery, store procedure, and user access control and administration, MySQL Workbench provides all the comprehensive tools such as data modeling, SQL development and much more. Thus, MySQL Workbench is a complete software package for designing and developing the database system. This tool is available for many operating systems such as Mac OS X, Windows, and Linux. Some of the features of MySQL Workbench are listed below:

- It provides the database administration features such as local and remote server control, setting up the MySQL server, and the server diagnostic information, including network traffic, server status, memory consumption, and much more.
- For security reason, it provides SSH-Tunneling, SSL, and Authorization and authentication connection management system to connect several servers locally as well as remotely.
- It has a user and session management system, which provides the special features for creating, deleting, viewing MySQL users and controlling the user privilege. For the project, MySQL Workbench community 6.2.5 was downloaded and connected to the remote database server (Raspberry Pi) to design the database as desired one and monitor it. Below is the screenshot of connected server using MySQL Workbench.

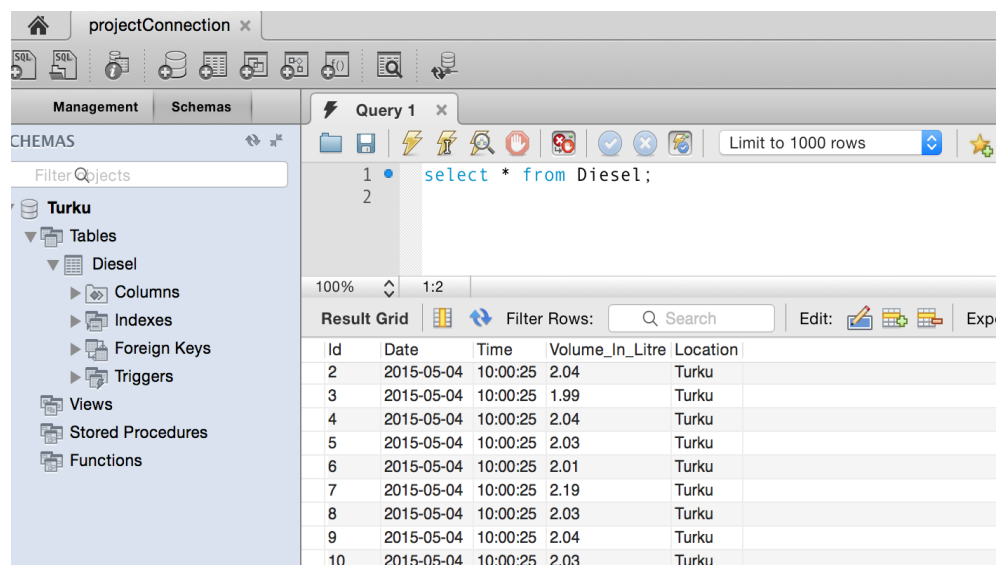


Figure 23. Screenshot of Database management in MySQL Workbench


6.5.2 Secure Remote Connection

To establish the secure network connection with the remote destination server, the author used some network methodologies, for example, SSH and VNC. Every remote

Raspberry Pi should have SSH and VNC software tools to have a secure connection management.

6.5.3 Web Interface

For user convenience, the web interface was created by using ASP.NET and C#. The interface displays the data (level of liquid) on the different locations. As web page is accessible over the Internet, therefore, the access should be restricted to the authorized user. To authenticate the user, the user information's must be stored in the database system or to an authentication key (password) must be provided to the user to get access to the website. Figure 24 displays the current liquid level in the tank.



Id	Date	Time	Volume_In_Litre	location
1	22.4.2015 0:00:00	13:28:12	0,42	Turku
2	22.4.2015 0:00:00	13:30:58	0,05	Turku
3	22.4.2015 0:00:00	13:33:24	0,05	Turku
4	22.4.2015 0:00:00	13:35:57	0,11	Turku
5	25.4.2015 0:00:00	20:15:21	1,72	Turku
6	25.4.2015 0:00:00	20:30:04	1,671	Turku
7	25.4.2015 0:00:00	21:00:02	1,73	Turku
8	25.4.2015 0:00:00	21:30:01	1,72	Turku
9	25.4.2015 0:00:00	21:30:05	1,671	Turku
10	25.4.2015 0:00:00	22:00:01	1,75	Turku
11	25.4.2015 0:00:00	22:00:04	1,73	Turku

Figure 24. Screenshot of data display on the web page

7 DATA SECURITY AND CHALLENGES

In the section, the challenges that the author has faced during the development of the project and its security measures are discussed.

7.1 Network and Data Security

In public and private networks, the security and robustness of any connection seems as a challenging topic, which definitely requires serious attention while developing network architecture. In most of the cases, SSH and VPNs are two different network methodologies implemented to establish a hidden connection to access a remote destination in order to accomplish the desired task. Nevertheless, it is never 100% secure and safe. In a similar manner, the network adopted in this project is established through VNC, which is popular, nowadays, especially for remote management. Another challenge is also data security.

The following approaches have been adopted to strengthen data security and network integrity:

- Authorization and authentication play a vital role in limiting the data access to the users. Primarily, they hide the data from ineligible users in order to boost the security level.
- Object privilege and system privilege aid in strengthening the security level in the system architecture.
- Remote connection is achieved through a secure socket layer connection like SSL, SSH or VPNs, via cellular or DSL or modem networks.
- Client software tools implement encryption and security protocol to secure administrator and user access.

- The security chain varies in the system chronology as the user access level and complexity of system increases in order to provide 24/7 uninterrupted accesses to high level of customer satisfaction and zero level networks down.
- Since various data and network threats are increasing day by day, the system could be upgraded with advancing technology developed for Raspberry Pi. For example, intruders could be fooled by changing known port numbers to anonymous ports, IP tables could be modified, open source could be given priority to increase security modification and so on.

7.2 Challenges

During the development of the project, the different challenges were encountered.

- Since Raspberry Pi does not have a built-in hard disk, the external SD card is used for storing the data and loading the configuration files. Thus, a SD card below 8 GB may be problematic because it easily becomes full and stops the running process. Apart from this, the data may become corrupted if the power source of the Raspberry Pi fails to supply and is disconnected without the proper shutdown of Raspberry Pi. Therefore, to avoid such difficulties, a SD card of 8 GB or above should be used and backup power source should be managed in case of power source failure.
- As the whole system depend upon the Internet connection, an uninterrupted Internet connection is needed for the proper functionality of the work. While using the Internet through a router, it is necessary to forward the specific port of router to gain access to the remote destination address because the default configuration of router blocks unauthorized access. Raspberry Pi must have a static IP address to be connected to the central station.

8 CONCLUSION

Although Raspberry Pi is developed with the intention for teaching computer science to the school level students, the use of Raspberry Pi with the combination of embedded hardware devices gives birth to modern technology systems, for example, remote connections, home-made media centers, remote video surveillance, Pi in the sky, which make our day-to-day life more convenient.

The main purpose of this thesis was to track and monitor the liquid levels remotely. With the combination of low cost hardware devices such as Ultrasonic Sensor, Arduino, Raspberry Pi along with the software tools, for example, MySQL Workbench, ASP.NET, Python and much more the desired goal was achieved. This project can be applied as real life solutions like in a fuel station, to manage daily fuel level (storage) in the station, or in any other remote liquid level management areas, where liquid level notification is needed, for example, daily water level notification from a water provider. Thus, this solution has a wide range of application and is very economical.

REFERENCES

- Arduino and Raspberry Pi Communication. 6 August 2014. [Online] Available at: <https://www.youtube.com/watch?v=eYu4vHDxA3I>. [Accessed: 25 March 2015]
- Arduino Uno. [Online] 2015 Available at: <http://docs-arduino.cc/en/main/arduinoBoardUno>. [Accessed: 15 February 2015]
- Attenuation of Sound waves. [Online] Available at: <https://www.nde-ed.org/EducationResources/CommunityCollege/Ultrasonics/Physics/attenuation.htm>. [Accessed: 4 March 2015]
- Calculator Soup, Tank Volume and Fill Calculator. [Online] Available at: <http://www.calculatorsoup.com/calculators/construction/tank.php>, and <http://www.mathopenref.com/segmentareaht.html>. [Accessed: 28 February 2015]
- Configuration of Arduino from Raspberry Pi. 2013. [Online] Available at: <http://razzpisampler.oreilly.com/ch10.html>. [Accessed: 5 March 2015]
- Configuration of Static IP address to Raspberry Pi. 19 July 2013. [Online] Available at: <http://www.modmypi.com/blog/tutorial-how-to-give-your-raspberry-pi-a-static-ip-address>. [Accessed: 5 March 2015]
- CSS tutorial – W3Schools, 2015. [Online] Available at: <http://www.w3schools.com/css/>. [Accessed: 24 March 2015]
- HTML and CSS – W3C, 2015. [Online] Available at: <http://www.w3.org/standards/webdesign/htmlcss#whatcss>. [Accessed: 20 March 2015]
- HTML Tutorial – W3Schools, 2015. [Online] Available at: <http://www.w3schools.com/html/default.asp>. [Accessed: 25 March 2015]
- Measuring the distance by HC-SR04 Ultrasonic Module. 15 February 2013. [Online] Available at: <https://www.youtube.com/watch?v=PG2VhpkPqoA>. [Accessed: 5 March 2015]
- Microsoft APS.NET, 2015. [Online] Available at: <http://www.asp.net/get-started/websites>. [Accessed: 24 March 2015]
- Microsoft Visual Studio, 2015.. [Online] Available at: <https://www.visualstudio.com>. [Accessed: 11 March 2015]
- MySQL Workbench 6.3.3 - MySQL . [Online] Available at: <http://dev.mysql.com/doc/workbench/en/wb-what-is-new-60.html/>. [Accessed: 12 March 2015]
- MySQL- Getting Started, 2015. [Online] Available at: <http://asp.net-tutorials.com/mysql/getting-started/>. [Accessed: 10 March 2015]
- Passive Ultrasonic Motion Sensor. 10 September 2011. [Online] Available at: <https://ultrasonicsensor.wordpress.com/>. [Accessed: 5 February 2015]
- Python - Python.org, 2015. [Online] Available at: <https://www.python.org/about/>. [Accessed: 15 March 2015]
- Raspberry Pi - Element 14, 2015. [Online] Available at: <http://downloads.element14.com/raspberryPi1.html>. [Accessed: 22 February 2015]

Raspberry Pi Module B - Allied Electronics 2015. [Online] Available at: <http://www.alliedelec.com/raspberry-pi-raspberry-pi-b-/70377493/>. [Accessed: 21 February 2015]

Raspberry Pi Module B, [Online] Available at: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>. [Accessed: 20 February 2015]

Raspbian – raspbian.org [Online]. Available at: <http://www.raspbian.org/>. [Accessed: 5 March 2015]

The Newton_Laplace Equation and Speed of Sound - Thermaxxjackets, 2015. [Online] Available at: <http://www.thermaxxjackets.com/newton-laplace-equation-sound-velocity/>. [Accessed: 5 February 2015]

Ultrasoni Principle – Micorsonic Ultrasonic Senosr, 2015. [Online] Available at: <http://www.microsonic.de/en/Interesting-facts.html>. [Accessed: 10 February 2015]

Ultrasonic Ranging Module HC-SR04. [Online] Available at: <http://www.micropik.com/PDF/HCSR04.pdf>, and http://www.electfreaks.com/store/download/product/Sensor/HCSR04/HCSR04_Ultrasonic_Module_User_Guide.pdf. [Accessed: 15 February 2015]

VNC, Port Configuration. [Online] Available at : <https://www.realvnc.com/products/vnc/documentation/5.0/guides/user/ai1047203.html>. [Accessed: 5 March 2015]

APPENDICES

Appendix 1.0 - source code to setup the Arduino for HCSR-04 Sensor

/* Connection between HC-SR04 sensor and Arduino Uno board:

VCC - Arduino >> 5V Pin

GND - Arduino >> GND pin

Echo - Arduino >> Pin 7

Trig - Arduino >> Pin 8 */

#define EchoPin 7 // for Echo pin

#define TrigPin 8 // for Trigger pin

double r =4.5; // radius of cylindrical vessel in centimeter(inner radius)

double h = 40 ; // length of vessel in centimeter (inner side)

int maxRange = 300; // Limitation for maximum range

int minRange = 0; // Limitation for minimum range

void setup()

{

Serial.begin (9600); // data communication rate

pinMode (TrigPin, OUTPUT); // set as output to TrigPin

pinMode (EchoPin, INPUT); // set as input to Echo Pin

}

void loop()

{

double distance, volume, duration, litre;

digitalWrite (TrigPin, HIGH); // set the Pin high

delayMicroseconds(10);

```

digitalWrite (TrigPin, LOW); // set the Pin low

delayMicroseconds(100);

duration = pulseIn(EchoPin, HIGH); // calculate time
distance = duration/58.2; // calculate distance in 'centimeter'
volume = 3.14 * pow(r,2) * (h-distance); // volume in centimeter cube (cm^3)
litre = volume/1000 ; // volume of liquid in litre unit

if(distance >= maxRange || distance <= minRange)
{
    Serial.println("Out of Range !!!"); //prints the given line if the sensor find the object beyond
the given limits(maxRangeand minRange value)

}
else
{
    Serial.println( litre);
}
delay (30000);

}

```

Appendix 2.0 – Python script source code to retrieve the data from the Arduino to Raspberry Pi

```

#!/usr/bin/python
import serial
import MySQLdb
import sys
import time
import datetime

# Connection to the database
databaseConn = MySQLdb.connect("localhost", "ganga", "gangapasswd", "Turku")
cursor = databaseConn.cursor() # open cursor to the databasse

```

try:

```

device = '/dev/ttyACM1' # serial connection port to the Arduino board
arduinoconnection = serial.Serial(device,9600) # communication rate
data = arduinoconnection.readline() # read the sensor data from the arduino

except Exception, ex:
    print "date: %s Time: %s Error in serial oppening port:" % (time.strftime("%Y-%m-%d"), time.strftime("%H:%M:%S")) +str(ex)
    sys.exit(1)
arduinoconnection.flushInput()
arduinoconnection.flushOutput()

while (True):

    location = 'Turku'
    date = time.strftime("%Y-%m-%d")
    clock= time.strftime("%H:%M:%S")
    cursor.execute("INSERT INTO Diesel(Date, Time, Volume_In_litre, Location) values(%s,%s,%s,%s)", (date, clock ,data, location))
    databaseConn.commit()
    time.sleep (3600) # stop for 1 hour

```

Appendix 3.0 – Source code of HTML file

```

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

    <title>WebPageGasStation</title>

    <link rel="stylesheet" href="Styles/StyleSheet.css" type="text/css"/>

    <asp:ContentPlaceHolder id="head" runat="server">

    </asp:ContentPlaceHolder>

</head>

<body>

    <form id="form1" runat="server">

    <div>

        <div id ="wrapper">

            <div id="banner">

                <asp:Image ID="Image1" runat="server" Height="380px" ImageUrl="~/Image/Gas-Stationcc.jpg" Width="100%" />

            </div>

```

```

<div id ="navigation">
    <ul id ="nav">
        <li><asp:HyperLink      ID="HyperLink1"      runat="server"
NavigateUrl="~/Pages/Station.aspx">GStation</asp:HyperLink></li>
        <li><asp:HyperLink      ID="HyperLink2"      runat="server"
NavigateUrl="~/Pages/About.aspx">About</asp:HyperLink></li>
        <li><asp:HyperLink      ID="HyperLink3"      runat="server"
NavigateUrl="~/Pages/Contact.aspx">Contact</asp:HyperLink></li>

    </ul>
</div>
<div id ="content">
    <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server" >

    </asp:ContentPlaceHolder>
</div>
</div>
</div>
</form>
</body>
</html>

```

Appendix 4.0 – Source Code of CSS file (Style.css)

```

body {
    color:#574c3f;
    font-family:Corbel, Arial, Helvetica, sans-serif, Verdana, sans-serif;
    font-size:16px;
    background-color: white;
}

```



```
#wrapper {  
    width:1100px;  
    margin:auto;  
    padding-bottom: 20px;  
    background-color:white;  
    moz-border-radius: 25px;  
    webkit-border-radius: 25px;  
    border-radius: 25px;  
    overflow: hidden;  
}
```

```
#banner{  
  
}
```

```
#navigation{  
    height: 45px;  
    background:linear-gradient(to top, #c2bcb5, #ffffff);  
}
```

```
#nav{  
    display: block;  
    list-style: none;  
    text-align: center;  
}
```

```
#nav ul{  
    display: none;  
    list-style: none;  
  
}
```

```
#nav li{
    float: left;
    display: block;
    margin-top: 15px;
    text-transform: uppercase;
    padding: 0px 20px 0px 20px;
    border-left: 1px solid #FFFFFF;
}

#nav a:link, #nav a:visited, #nav a:active{
    display: block;
    color: #574c3f;
    text-decoration: none;
}

#nav a:hover{
    background-color: aliceblue;
}

#nav a:visited {
    display: block;
    color: #574c3f;
    text-decoration: none;
}

#content{
    padding:20px;
    min-height: 400px;
}
```

```
/*GasStation */
```

```
#Content2{
    float: left;
    width: 250px;
    height: 250px;
    text-align: center;
}
```

```
.productImage{
    width :150px;
    height: 100px;
    vertical-align:center;
    padding: 50px;
    border-radius:50px;
    -moz-border-radius:50px;
    -webkit-border-radius:50px;
}
```

Appendix 5.0 - C# Source code to display data on the web page

```
public partial class Pages_Station : System.Web.UI.Page
{

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            Data_Connection();
            //CheckConnection();
        }
    }
}
```

```

protected void Data_Connection()
{
    String StringgConn =
    System.Configuration.ConfigurationManager.ConnectionStrings["GasStationConnString"].ToString(); //Connect to ConnectionStrings-"GasStationConnString" and store in StringgConn

    using (MySqlConnection ganga = new
    MySql.Data.MySqlClient.MySqlConnection(StringgConn)) // create new object
    {
        String qurstring = "Select * from Turku.Diesel";

        MySqlDataAdapter mala = new MySqlDataAdapter(qurstring, ganga);

        DataTable DTable = new DataTable();
        mala.Fill(DTable);

        if (DTable.Rows.Count > 0)
        {
            GridView1.DataSource = DTable;
            GridView1.DataBind();
        }
    }
}

```

Appendix 6.0 – Database Connection String

```
<connectionStrings>
```

```
  <add name ="GasStationConnString"
```

```
    connectionString="Server=192.168.11.7; UserId= ganga; Password=gangapasswd;
    Database=Turku;"
```

```
    providerName="Mysql.Data.MySqlClient"/>
```

```
</connectionStrings>
```

Appendix 7.0 – ASP.NET Source Code

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Station.aspx.cs" Inherits="Pages_Station" %>
```

```
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
```

```
    <asp:GridView ID="GridView1" runat="server" CaptionAlign="Left" CellPadding="4"
ForeColor="#333333" GridLines="None" HorizontalAlign="Center"
UseAccessibleHeader="False" Width="100%"
OnSelectedIndexChanged="GridView1_SelectedIndexChanged" DataKeyNames="Id">
```

```
    <AlternatingRowStyle BackColor="White" />
```

```
    <EditRowStyle BackColor="#2461BF" />
```

```
    <FooterStyle BackColor="#507CD1" Font-Bold="True" ForeColor="White" />
```

```
    <HeaderStyle BackColor="#507CD1" Font-Bold="True" ForeColor="White" />
```

```
    <PagerStyle BackColor="#2461BF" ForeColor="White" HorizontalAlign="Center" />
```

```
    <RowStyle BackColor="#EFF3FB" />
```

```
    <SelectedRowStyle BackColor="#D1DDF1" Font-Bold="True" ForeColor="#333333" />
```

```
    <SortedAscendingCellStyle BackColor="#F5F7FB" />
```

```
    <SortedAscendingHeaderStyle BackColor="#6D95E1" />
```

```
    <SortedDescendingCellStyle BackColor="#E9EBEF" />
```

```
    <SortedDescendingHeaderStyle BackColor="#4870BE" />
```

```
</asp:GridView>
```

```
</asp:Content>
```